

# Memory Log

*Watch for changes*

Tue, Oct 15, 2002

While debugging front end software, it is often interesting to watch memory via the Memory Dump page application. This note considers an idea of a local application that can be used to monitor changes in memory and log them.

The basic idea is to watch for a region of memory to change, and whenever it does, log the changed value and include the time-of-day the change was detected.

First, assume that only a 1-byte, 2-byte, or 4-byte region need be monitored for changes.

## *Parameter layout*

ENABLE	B	Enable Bit#
ID/SIZE		ID, Size of data monitored: 1,2,4
ADDR1	HI	Memory address #1
ADDR1	LO	
ADDR2	HI	Memory address #2
ADDR2	LO	
ADDR3	HI	Memory address #3
ADDR3	LO	
ADDR4	HI	Memory address #4
ADDR4	LO	

Any memory address can be entered, but address 0x00000000 is invalid. If the contents of any of the specified addresses changes value, a record is written to a log. As is typical, let the log be a data stream. The name used could be MEMLOGxx, where xx is specified as the hex ascii value of the ID parameter. This allows for more than one such instance of MLOG. The record format can be:

ID_Size	2	ID (hi byte) and size (lo byte)
Addr#	2	Specifies which address parameter, range 1-4
value	4	New data value, from 1-4 bytes, depending on SIZE parameter
time	8	Usual BCD time-of-day

Alternatives might allow specifying an LA name (4-characters) plus an offset. This would mean fewer addresses could be specified, but 4 addresses may be overkill.

Another alternative may be to expand the byte that indicates the size of what is to be monitored in order to permit monitoring a bit within a byte. To this end, use values such as 0xb0, where b is the bit# in the range 0-7. The address would point to the byte containing the bit to be monitored for changes. The only values logged in this case would be 0 or 1. The other three possibilities for this byte are entered as 0x01, 0x02, and 0x04, specifying a byte, a 16-bit word, or a 32-bit long word.

One could compare this with alarm logging. To monitor a 2-byte value, for example, one could add an entry to the Data Access Table to copy the value to a channel, then enable alarm scanning with a zero tolerance. But if one is monitoring a counter, say, one would also have to alter the nominal value every time it changed. Also, there is more hassle to change the Data Access Table and enter a new analog descriptor. In addition, it increases the alarm message clutter. This new scheme is much more informal.